ProtSynGen: Knowledge Graph-Guided Open-Ended Data Synthesis for Protein LLMs

Anonymous ACL submission

Abstract

The performance of large language models (LLMs) depends critically on the quality of their instruction-tuning data. For protein LLMs, existing instruction-tuning datasets are typically derived from structured protein annotations through template-based or label-guided generation. Although these datasets capture important biological facts, their construction paradigms often lead to two limitations: instruction homogenization and reasoning superficiality. The former refers to questions concentrating around repetitive forms and narrow topics; the latter to answers relying on shallow annotation lookup rather than evidence-rich biological reasoning. To address these limitations, we propose ProtSynGen, a knowledge graph-guided open-ended data synthesis framework for protein LLMs. ProtSynGen integrates structured protein databases with unstructured scientific literature into a protein-centric knowledge graph, and uses hierarchical graph sampling to guide LLMs in generating open-ended QAs across atomic, aggregated, and complex reasoning scenarios. Experiments show that ProtSynGen produces instruction-tuning data with broader topic coverage, richer reasoning depth, and stronger quality metrics than existing protein LLM instruction datasets. Models trained on ProtSynGen data further achieve superior performance on open-ended protein QA benchmarks, consistently outperforming baseline methods across atomic, aggregated, and complex reasoning scenarios.

1 Introduction

Instruction-tuning has become a standard approach for adapting large language models (LLMs) to diverse instruction-following scenarios, but its effectiveness depends critically on the quality of the underlying instruction data (OpenAI, 2023; DeepSeek-AI et al., 2025; Yang et al., 2025). Recent studies have extended instruction-tuning to the protein domain, aiming to develop protein LLMs that can process protein sequences alongside certain natural language instructions (Xiao et al., 2024; Guo et al., 2023; Luo et al., 2026; Wang et al., 2024; Carrami and Sharifzadeh, 2024; Wang et al., 2025a; Zhou et al., 2025). These models are trained on instruction-tuning datasets to perform tasks such as analyzing protein sequences, answering questions about protein functions, and reasoning about biological interactions.

As shown in Figure 1, existing instruction-tuning datasets for protein LLMs are mostly constructed by two methods: **template-based generation** and **label-guided generation**. The former relies on manually designed question-answer templates with protein annotations, while the latter uses annotation labels to prompt LLMs to generate QA pairs. For example, OPI (Xiao et al., 2024) and BioMedGPT (Luo et al., 2026) use predefined templates for topics such as protein names, functional annotations, and family classification; Pika (Carrami and Sharifzadeh, 2024), in contrast, follows label-guided generation with labels like molecular mass, EC number, and subcellular localization. While these datasets enable models to perform several protein-related tasks, their dependence on predefined templates, LLM prompts, and structured annotations introduces two fundamental limitations:

(1) Instruction homogenization: Because existing datasets are anchored in predefined templates or limited label sets, their questions often concentrate around similar forms, topics, and difficulty levels. This homogenization can make models overly adapted to narrow instruction patterns in training datasets, limiting their ability to handle diverse open-ended protein questions.

(2) Reasoning superficiality: Existing datasets primarily rely on protein-to-label annotations. As a result, their answers often reduce protein reasoning to short mappings from a protein to an annotation label. However, open-ended protein

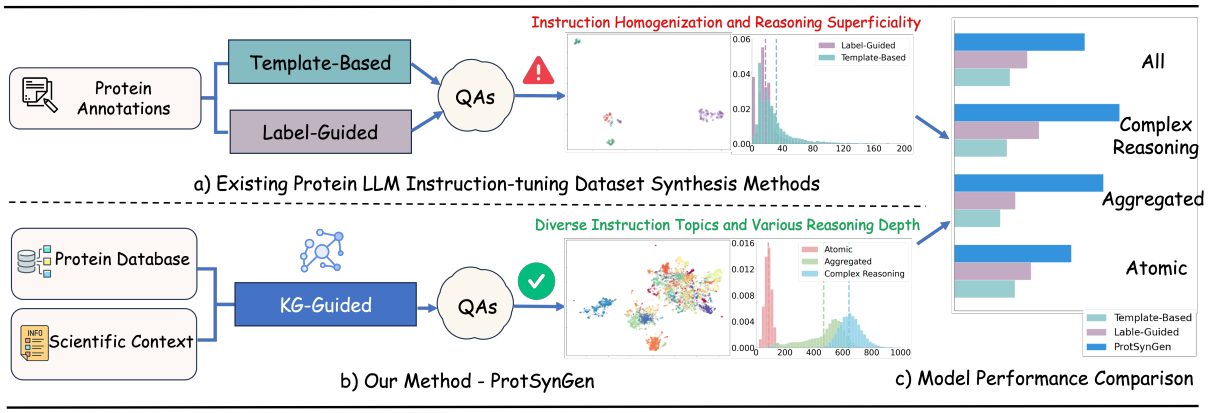

Figure 1: Overview of the motivation of ProtSynGen. (a) Existing protein LLM instruction-tuning datasets rely on template-based or label-guided generation, leading to homogeneous instructions and shallow annotation-based reasoning. (b) ProtSynGen integrates structured protein databases with scientific literature and performs knowledge graph-guided hierarchical sampling to synthesize diverse open-ended QAs with multi-level reasoning. (c) Models trained on ProtSynGen data outperform baselines on open-ended protein QA benchmarks across atomic, aggregated, and complex reasoning scenarios.

QA frequently requires connecting database facts with literature evidence, functional mechanisms, experimental context, and related biological entities. Models trained on shallow annotation-style data may therefore struggle to produce coherent and well-supported scientific answers.

These limitations motivate us to propose a data synthesis framework that improves both instruction diversity and reasoning depth. To this end, we propose ProtSynGen, a knowledge graph-guided open-ended data synthesis framework for protein LLMs. We integrate protein databases with scientific literature into a protein-centric knowledge graph. Based on this graph, we use hierarchical graph sampling to guide LLMs in synthesizing QA pairs across different scenarios. Specifically, as shown in Figure 2, ProtSynGen consists of four stages: (i) knowledge graph establishment from databases; (ii) knowledge graph enrichment guided by scientific literature; (iii) hierarchical graph sampling for data synthesis; and (iv) fact checking and quality audit. Following this pipeline, we curate ProtSynGen-SFT, a protein instruction-tuning dataset containing 5.16 million open-ended protein QA pairs covering 483,954 protein sequences.

Experiments show that ProtSynGen-SFT exhibits substantially higher instruction uniqueness, broader semantic coverage, and a more balanced topic distribution than existing protein LLM instruction data. Moreover, models trained on ProtSynGen-SFT data consistently outperform baselines on open-ended protein QA benchmarks across atomic, aggregated, and complex reasoning scenarios.

The contributions of this paper are:

- We propose ProtSynGen, a knowledge graph-guided open-ended data synthesis framework for protein LLMs which integrates protein databases with scientific literature into a protein-centric knowledge graph.
- We curate ProtSynGen-SFT, an open-ended protein QA dataset containing 5.16 million QA pairs, covering atomic, aggregated, and complex reasoning scenarios.
- Experiments show that ProtSynGen yields higher-quality synthetic instruction data and improves the open-ended protein QA performance of protein LLMs.

2 Methodology

2.1 Knowledge Graph Establishment from Databases

Protein databases (e.g., SwissProt) provide manually reviewed, schema-consistent annotations for proteins, but these annotations are often treated as independent fields in existing data construction pipelines. ProtSynGen instead organizes them as a basic protein-centric knowledge graph, so that protein facts can be linked, sampled, and audited as structured evidence. We define the basic graph as:

$$G^{\text{basic}} = (V^{\text{basic}}, E^{\text{basic}}), \quad (1)$$

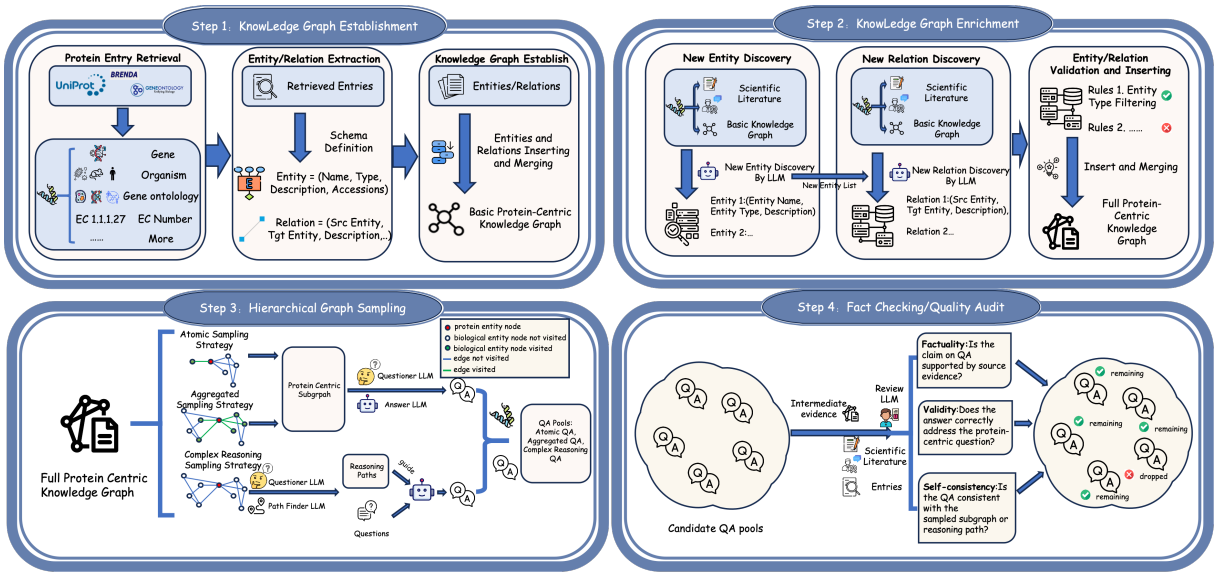

Figure 2: ProtSynGen consists of four stages: (a) knowledge graph establishment from databases, (b) knowledge graph enrichment guided by scientific literature, (c) hierarchical graph sampling for data synthesis, and (d) fact checking and quality audit.

where V^{basic} denotes biological entity nodes, including proteins and database annotation entities such as GO terms, enzyme classes, organisms, genes, and sequence features, and E^{basic} denotes typed relations between them. Each node $v \in V^{\text{basic}}$ is defined as

$$v = (n, t, d, src), \tag{2}$$

where n is the standardized entity name, t is the entity type defined by the database schema, d is a textual description, and src records the source protein accessions. Each directed edge $e \in E^{\text{basic}}$ is defined as

$$e = (v_s, v_t, \rho, d_r, src), \tag{3}$$

where v_s and v_t are the source and target nodes, ρ is the relation type, d_r is the relation description, and src records supporting protein accessions.

To construct G^{basic}, we formalize the protein database $\mathcal{D}$ as a collection of entries indexed by protein accessions,

$$\begin{aligned} \mathcal{D} &= \{(u, \mathcal{R}(u)) \mid u \in \mathcal{U}\}, \\ \mathcal{R}_{\mathcal{S}}(u) &= \{r \in \mathcal{R}(u) \mid \text{type}(r) \in \mathcal{S}\}. \end{aligned} \tag{4}$$

where u denotes a protein accession, $\mathcal{R}(u)$ denotes its original annotation attributes, and $\mathcal{S}$ is the set of attribute types used for graph construction. For each accession u, ProtSynGen maps the protein to a protein node v_u and maps each valid annotation $r \in \mathcal{R}_{\mathcal{S}}(u)$ to an attribute node v_r. It then constructs two types of edges: protein-attribute edges $e^{(p)}_{u,r} = (v_u, v_r, \rho_{u,r}, d_{u,r}, u)$, which connect the protein node to each annotation node, and attribute-attribute edges $e^{(a)}_{r_i,r_j} = (v_{r_i}, v_{r_j}, \rho_{r_i,r_j}, d_{r_i,r_j}, u)$, which connect semantically dependent annotations within the same protein entry. This design preserves both direct protein facts and local relations among annotations. Specifically, by iterating over all accessions in SwissProt releases up to 2024-04 and applying entity disambiguation, cross-entry linking, and semantically consistent merging, ProtSynGen yields G^{basic}, which serves as the factual backbone for later knowledge graph enrichment with literature and data synthesis.

2.2 Knowledge Graph Enrichment Guided by Scientific Literature

Although G^{basic} provides high-precision database evidence, it is bounded by schema-defined annotations. Scientific literature complements this structured evidence with mechanistic findings, experimental context, and biological relations that are often absent from database fields. ProtSynGen therefore enriches the graph with literature-derived entities and relations, providing additional evidence that can support broader topic coverage and deeper biological reasoning, especially for complex reasoning scenarios.

For each accession u, let $C_u = \{c_u^k\}_{k=1}^{m_u}$ denote its literature context, including UniProt expert comments and associated PubMed abstracts. We define $G_u^{\text{basic}} = (V_u^{\text{basic}}, E_u^{\text{basic}}) \subseteq G^{\text{basic}}$ as the connected

protein-centered subgraph containing v_u and all entities and relations associated with u. During enrichment, ProtSynGen uses G_u^{basic} as structured grounding context and restricts generated entities to a predefined type set $\mathcal{T}$. This ensures newly introduced entities and relations remain connected to the existing protein-centered graph while staying consistent with the graph schema and avoiding duplication with already recorded entities. The enrichment pipeline then proceeds in three stages: entity discovery, relation discovery, and filtering and insertion:

New entity discovery: Given V_u^{basic}, C_u, and the entity discovery prompt I_e, an LLM $\mathcal{L}$ proposes literature-supported entities V_u^{new} that are absent from the current subgraph but conform to $\mathcal{T}$:

$$V_u^{\text{new}} \leftarrow \mathcal{L}(V_u^{\text{basic}}, C_u \mid I_e, \mathcal{T}). \tag{5}$$

New relation discovery: The relation discovery prompt I_r then asks $\mathcal{L}$ to generate literature-supported relations. To avoid over-centralizing newly added relations around the protein node, ProtSynGen separates intrinsic relations among newly discovered entities from extrinsic relations between new and existing entities:

$$\begin{aligned} E_u^{\text{in}} &\leftarrow \mathcal{L}(V_u^{\text{new}}, C_u \mid I_r), \\ E_u^{\text{ex}} &\leftarrow \mathcal{L}(V_u^{\text{new}}, V_u^{\text{basic}}, C_u \mid I_r). \end{aligned} \tag{6}$$

Filtering and insertion: The candidate entities V_u^{new} and relations $E_u^{\text{new}} = E_u^{\text{in}} \cup E_u^{\text{ex}}$ are filtered by type validity and connectivity to G_u^{basic}. The remaining entities and relations are normalized and inserted using the same node-edge schema defined in Section 2.1.

After processing all accessions, we obtain the enriched protein-centric knowledge graph $G^{\text{full}} = (V^{\text{full}}, E^{\text{full}})$, which expands the evidence space in both breadth and depth for subsequent graph sampling.

2.3 Hierarchical Graph Sampling for Data Synthesis

The enriched graph G^{full} provides a broad evidence space for protein-centric QA synthesis. To convert this graph into controllable generation contexts, ProtSynGen samples protein-centered subgraphs rather than using the full graph indiscriminately. For each accession u, let $G_u^{\text{full}} \subseteq G^{\text{full}}$ denote the connected enriched protein-centered subgraph associated with u. The breadth of a sampled subgraph affects the range of biological topics involved in a question, while the depth affects the reasoning complexity required for the answer. We define three sampling scenarios with progressively increasing reasoning complexity:

Atomic: This scenario targets a single one-hop edge adjacent to the protein node v_u, supporting a QA pair about one discrete protein fact.

Aggregated: This scenario targets a connected subgraph containing v_u, supporting a QA pair that requires integrating multiple annotations or relations.

Complex reasoning: This scenario targets application-oriented QAs that require integrating multiple biological entities, relations, and contextual evidence within a protein-centered subgraph.

Formally, for each accession u, ProtSynGen samples multiple subgraphs from the enriched protein-centered subgraph G_u^{full} according to a chosen scenario r. We denote the j-th sampled subgraph under scenario r as g_u^j, drawn by a scenario-specific sampler $\mathcal{S}_r$:

$$\begin{aligned} &g_u^j \sim \mathcal{S}_r(G_u^{\text{full}}), \\ &r \in \{\text{atomic}, \text{aggregated}, \text{complex}\}. \end{aligned} \tag{7}$$

For atomic and aggregated scenarios, ProtSynGen generates a question and answer conditioned on the sampled subgraph g_u^j:

$$Q_u^j \leftarrow \mathcal{L}_q(g_u^j \mid I_q), \quad A_u^j \leftarrow \mathcal{L}_a(g_u^j, Q_u^j \mid I_a). \tag{8}$$

For complex reasoning, ProtSynGen first generates a graph-conditioned question from G_u^{full}, and then uses a Path Finder LLM to derive an explicit reasoning path P_u^j from G_u^{full} for answering the question. The answer is then generated from this path:

$$\begin{aligned} Q_u^j &\leftarrow \mathcal{L}_q(G_u^{\text{full}} \mid I_q), \\ P_u^j &\leftarrow \mathcal{L}_p(G_u^{\text{full}}, Q_u^j \mid I_p), \\ A_u^j &\leftarrow \mathcal{L}_a(P_u^j, Q_u^j \mid I_a). \end{aligned} \tag{9}$$

Prompts are tailored to each reasoning scenario. Atomic and aggregated samples are generated in a fixed ratio, and visited edges are tracked to reduce redundant evidence reuse. Complex reasoning is enabled only when the diameter of G_u^{full} exceeds a predefined threshold. After hierarchical sampling and QA generation, the resulting synthetic dataset is:

$$D_{\text{synth}} = \bigcup_{u \in \mathcal{U}} \left\{ \left(u, Q_u^j, A_u^j, r_u^j\right) \right\}_{j \in [M_u]}. \tag{10}$$

2.4 Fact Checking and Quality Audit

The synthetic dataset D_{synth} may still contain unsupported or inconsistent statements in QAs. For each candidate $x = (u, Q_u^j, A_u^j, r_u^j)$, ProtSynGen assembles two types of evidence: original contexts including database annotations $\mathcal{R}(u)$ and literature context C_u, and intermediate structured evidence including the sampled subgraph g_u^j or reasoning path P_u^j. An LLM-based reviewer audits x for **factivity** with respect to the original context, **validity** of the answer to the question, and **self-consistency** between the QA pair and the intermediate structured evidence. Candidates failing any criterion are removed:

$$D_{\text{final}} = \{x \in D_{\text{synth}} : \text{Audit}(x) = 1\}. \quad (11)$$

The final filtered dataset D_{final} is released as ProtSynGen-SFT, which contains 5.16 million open-ended protein QA pairs.

We validate the effectiveness of this stage through **manual verification** on 15 randomly sampled proteins, auditing all 979 edges in their knowledge subgraphs and 100 generated QA pairs against source SwissProt entries and cited PubMed abstracts. Expert verification in this sampling subset shows that 85.7% of sampled edges are valid. Furthermore, the fact checking and quality audit stage improves the expert-validated QA precision from 67% to 83.8% by operating with 100% reject precision in the 20 rejected QA pairs, meaning every filtered item is confirmed problematic by expert review. These results support the reliability of the generation pipeline and suggest that the audit stage effectively removes residual errors in our sampled audit set. Detailed implementation of ProtSynGen is provided in Appendix A.3, and manual validation results are reported in Appendix A.4.

3 Analysis of Synthesized Instruction Data

3.1 Overview Statistics

We evaluate ProtSynGen-SFT from three perspectives: instruction diversity, reasoning depth, and data quality. As a corpus-level overview, Table 1 summarizes dataset scale, construction format, distinct-instruction ratio, and average response length; a fuller statistical table is provided in Appendix A.1.

Compared with template-based and label-guided baselines, ProtSynGen-SFT combines substantially higher instruction uniqueness with longer responses, motivating the more targeted analyses below.

Dataset	#Samples	Distinct Inst. (%)	Avg Resp. Len	Format
SwissProt-V3	430K	–	75.80	template-based
ProteinKG25	422K	–	141.45	template-based
Mol-Instruction	299K	0.020	59.32	template-based
OPI	1.78M	0.009	31.87	template-based
UniProtQA	1.51M	0.000	25.85	template-based
PDBQA	3.36M	0.001	3.90	template-based
Pika-DS	1.71M	5.671	17.81	label-guided
ProtSynGen-SFT	5.16M	80.229	179.11	KG-guided

Table 1: Overview statistics of protein LLM datasets. Distinct Inst. denotes the percentage of distinct instructions in the dataset, and Avg Resp. Len denotes average response length computed using the SapBERT-PubMedBERT tokenizer (Liu et al., 2021). "–" indicates that the dataset does not contain explicit natural-language instructions. SwissProt-V3, ProteinKG25, and PDBQA are released with ProteinChat (Guo et al., 2023).

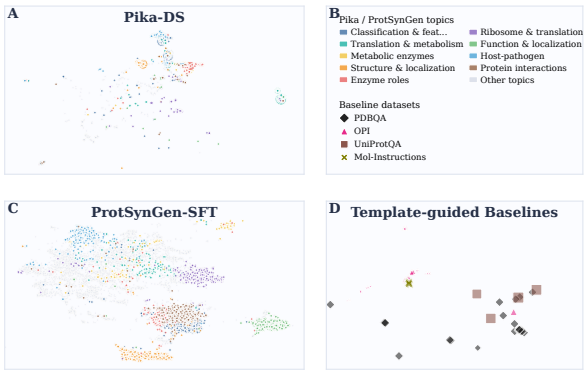

Figure 3: UMAP visualization of instruction embeddings across datasets. Instructions are encoded by SapBERT-PubMedBERT; template-based datasets use all available instructions, while Pika-DS and ProtSynGen-SFT are sampled with 10K instructions.

3.2 Instruction Diversity and Topic Coverage

We first examine the semantic coverage of instructions using SapBERT-PubMedBERT embeddings and UMAP. As shown in Figure 3, template-based datasets occupy a few narrow and isolated regions because they repeatedly reuse a small set of question templates. Pika-DS expands this space relative to template-based baselines, but its instructions remain concentrated around several dominant themes. In contrast, ProtSynGen-SFT spans a broader and more continuous semantic space, consistent with the high distinct-instruction ratio in Table 1.

We then quantify topic concentration through 50-topic clustering of 10K instruction samples from Pika-DS and ProtSynGen-SFT. As shown in Figure 4, the top five topics account for nearly 90% of Pika-DS, whereas they account for only about 32% of ProtSynGen-SFT. ProtSynGen-SFT covers

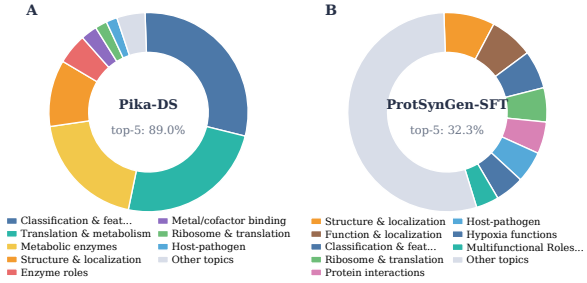

Figure 4: Topic composition of 10K sampled instructions from Pika-DS and ProtSynGen-SFT. Topic labels are assigned after clustering instruction embeddings into 50 clusters.

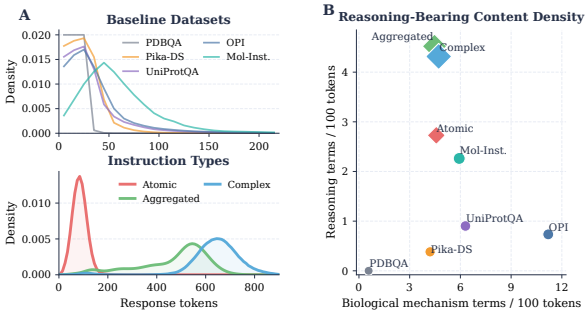

Figure 5: Response length and proxy reasoning-depth indicators across protein LLM instruction-tuning datasets. Panel (a) on the left compares response token length distributions, while Panel (b) reports normalized densities of reasoning terms and biological mechanism terms as estimates of factual reasoning.

the major Pika-DS topics while also introducing additional biological themes. The full 50-topic comparison and implementation details are provided in Appendix A.6. Together, these qualitative and quantitative views indicate that ProtSynGen reduces repetitive instruction patterns and broadens topic coverage beyond label-guided generation.

3.3 Response Length and Reasoning Depth

We next examine whether the broader semantic coverage of ProtSynGen-SFT is accompanied by richer response content rather than short annotation-style supervision. Following prior text-analysis work that uses discourse and lexical indicators to characterize explanatory structure in text (Graesser et al., 2004; Pennebaker et al., 2015), Figure 5 compares response length distributions and two lexical proxy indicators: reasoning terms, which approximate explicit causal, evidential, and inferential links between statements, and biological mechanism terms, which approximate protein-related mechanism content. These indicators are coarse diagnostics rather than direct measures of factual biological reasoning; full term lists, computation details, and caveats are provided in Appendix A.7.

As shown in Figure 5, many existing datasets concentrate on short responses, whereas ProtSynGen-SFT contains a broader distribution with substantially more long-form answers. In this two-dimensional lexical-indicator space, ProtSynGen-SFT is positioned higher on the reasoning-term axis while retaining biological mechanism terms, suggesting that knowledge graph-guided synthesis more often produces responses with explicit explanatory structure instead of only returning short annotation labels.

3.4 Data-Quality Evaluation

Dataset	MTLD ↑	Nat. ↑	Coh. ↑	Und. ↑	Reward ↑
Mol-Instruction	58.07	74.72	99.86	74.71	55.39
OPI	21.94	60.11	67.62	58.88	37.08
UniProtQA	15.67	46.64	92.75	46.05	24.67
PDBQA	1.03	33.96	22.90	31.10	17.26
Pika-DS	9.21	79.30	83.69	77.49	26.97
ProtSynGen-SFT	**111.84**	**84.78**	**99.90**	**87.34**	**69.21**

Table 2: Data-quality evaluation across datasets. Nat., Coh., and Und. denote UniEval naturalness, coherence, and understandability. Reward denotes Reward_Ind, a reward model that evaluates the quality of instruction-tuning data. All metrics except MTLD are reported on a 0–100 scale.

Finally, we test whether the gains in diversity and response depth come at the cost of data quality. Table 2 reports evaluations using MTLD (McCarthy and Jarvis, 2010), UniEval (Zhong et al., 2022), and Reward_Ind (Shi et al., 2024), with metric definitions and implementation details provided in Appendix A.8. ProtSynGen-SFT achieves the best scores on all reported metrics, suggesting that its broader coverage and longer responses do not simply reflect noisy verbosity, but are accompanied by stronger coherence, naturalness, and overall instruction-data quality.

4 Downstream Effectiveness of ProtSynGen

In this section, we assess whether diverse instructions and reasoning-oriented responses generated by ProtSynGen could improve open-ended protein QA performance. We evaluate models on two complementary benchmarks: LiveProtQA, a temporal benchmark over held-out SwissProt releases, and ExpertProtQA, an externally sourced human-scored benchmark. We compare ProtSynGen-SFT against publicly released protein and scientific LLMs as well as controlled retraining baselines,

and ablate key pipeline components to verify their contributions.

4.1 Experimental Setup

Training and Baselines. Comparing directly against published protein LLMs is insufficient to attribute gains to training data, as these models differ in base architecture, model scale, and training recipe. To enable a fair comparison, we report baseline models' performance from their publicly released checkpoints (ProtT3, Mol-Instruction, OPI-Galactica, BioMedGPT) and also fine-tune the same Qwen2.5-7B-Instruct backbone with their datasets under identical training configurations. Furthermore, existing SFT datasets vary considerably in scale across QA and protein sequences; we therefore use ProtSynGen-SFT-106K (a 106K-QA subset covering 10K proteins), which is the smallest among all compared datasets in both token count and sequence count (Appendix A.10), reducing the possibility that observed gains are driven by training scale. We also report the base Qwen2.5-7B-Instruct without protein-domain SFT as a baseline. Full training settings are in Appendix A.9.

Evaluation Benchmarks and Metrics. LiveProtQA is a temporal out-of-distribution benchmark constructed from four SwissProt releases held out after the ProtSynGen-SFT training cutoff, spanning SwissProt 2024_06 to 2025_03. After sequence-level homology filtering against ProtSynGen-SFT-106K, the benchmark covers only unseen proteins across atomic, aggregated, and complex reasoning scenarios. Model outputs are scored pointwise on a 1–5 scale by three LLM judges (GPT-4o, DeepSeek-R1, and Qwen3-235B) based on helpfulness and factivity relative to the protein entry and supporting literature; we report the mean judge score normalized to 0–100 and macro-averaged over the three scenarios. ExpertProtQA provides a complementary external evaluation using 118 open-ended questions adapted from public protein datasets and scored by domain experts. While LiveProtQA emphasizes temporal generalization and controlled reasoning scenarios, ExpertProtQA tests robustness to externally sourced question styles under expert evaluation. We use expert scores as the primary metric and additionally report LLM-judge scores as a secondary reference. For both benchmarks, we remove proteins with at least 80% sequence identity to the training set ProtSynGen-SFT-106K to reduce potential sequence-level leakage. More details are in Appendix A.12 and A.13.

4.2 Results on LiveProtQA

Table 3 reports results under two comparison protocols: publicly released checkpoints and retraining with a shared Qwen2.5-7B-Instruct backbone. ProtSynGen-SFT-106K achieves the highest average score in both settings. In the retraining block, where all models share the same backbone and training configuration, our model reaches an average score of 46.01, improving over the strongest retraining baseline (Mol-Instruction, 35.12) by 10.89 points.

The advantage also increases with reasoning complexity: our model improves over the respective runner-up by 4.84 on atomic, 10.51 on aggregated, and 16.59 on complex reasoning. This pattern is consistent with ProtSynGen's hierarchical graph sampling design, which exposes the model to QA pairs grounded in protein-centered subgraphs with progressively increasing reasoning complexity.

Beyond matched-backbone retraining, the ProtSynGen model also outperforms all publicly released protein and scientific LLMs in Table 3, including Intern-S1 (34.15), a 235B-parameter MoE model with a 6B encoder. These results suggest that ProtSynGen-SFT provides effective instruction-tuning data for open-ended protein QA, with gains visible both under controlled retraining and against strong released models.

4.3 Expert-Scored Evaluation on ExpertProtQA

ExpertProtQA tests whether the improvements hold under expert evaluation on externally sourced open-ended protein QA. As shown in Table 4, ProtSynGen-SFT-106K achieves an Expert Judge score of 48.09, substantially exceeding Intern-S1 (30.72), Mol-Instruction (26.06), Pika-DS (24.15), and UniProtQA (23.73). The LLM Judge scores preserve the same model ranking (ProtSynGen-SFT-106K at 49.36), and the two evaluation protocols show high agreement overall (within-1 agreement 95.9%, pairwise direction agreement 89.4%; full analysis in Appendix A.14). These findings suggest that the improvements from ProtSynGen-SFT generalize to externally sourced open-ended protein QA under expert evaluation.

Model Source	Model/Dataset	LiveProtQA Score ↑			
		Atomic	Aggregated	Complex	Avg.
Publicly released protein/scientific LLM checkpoints					
Public Release	InstructProtein	0.41	0.08	0.17	0.22
	ProtT3	3.75	0.52	0.43	1.57
	OPI-Galactica	12.49	8.56	8.75	9.94
	Mol-Instruction	14.21	9.15	16.01	13.12
	BioMedGPT	18.27	16.05	20.21	18.18
	Intern-S1	29.92	32.26	40.26	34.15
Retrained with the same architecture and different datasets					
Retrain	Qwen2.5-7B-Instruct	20.92	26.76	37.29	28.32
	PDBQA	6.09	2.98	1.32	3.46
	OPI	23.65	18.34	20.79	20.93
	UniProtQA	24.29	21.01	25.91	23.73
	Pika-DS	31.29	25.06	36.39	30.91
	Mol-Instruction	33.45	32.37	39.52	35.12
	ProtSynGen-SFT-106K	**38.29**	**42.88**	**56.85**	**46.01**

Table 3: Main results on LiveProtQA. Scores are normalized to 0–100. Avg. is the macro average over atomic, aggregated, and complex reasoning scenarios. The retraining block uses the same Qwen2.5-7B-Instruct backbone and changes only the SFT data.

Model/Dataset	Expert Judge ↑	LLM Judge ↑
UniProtQA	23.73	31.07
Pika-DS	24.15	29.38
Mol-Instruction	26.06	33.76
Intern-S1	30.72	34.82
ProtSynGen-SFT-106K	**48.09**	**49.36**

Table 4: ExpertProtQA results. Scores are normalized to 0–100. Expert Judge is the primary metric; LLM Judge scores are averaged across three judges.

Variant	LiveProtQA Score ↑			
	Atomic	Agg.	Complex	Avg.
Full	**38.29**	**42.88**	**56.85**	**46.01**
w/o KG Enrichment	37.08	41.65	54.21	44.31
w/o Quality Audit	37.15	42.00	54.04	44.40

Table 5: Ablation results on LiveProtQA. Scores are normalized to 0–100; Avg. averages the three reasoning scenarios.

4.4 Ablation Study

Finally, we ablate two components of ProtSynGen: literature-based KG enrichment and quality auditing. Each ablated dataset is generated from the same 10K protein subset and used to fine-tune the same backbone under identical settings. As shown in Table 5, removing either component leads to consistent score drops on LiveProtQA ($p < 0.001$, paired Wilcoxon; $n = 1{,}800$), with bootstrap 95% confidence intervals excluding zero. Similar drops are observed on ExpertProtQA ($p < 0.05$; $n = 118$; see Appendix A.15 for details). These results confirm that both KG enrichment and quality auditing contribute positively to the final model performance.

5 Conclusion

In this paper, we present ProtSynGen, a knowledge graph-guided open-ended data synthesis framework for protein LLMs. The key insight is that open-ended protein QA requires evidence grounded in connected biological knowledge rather than isolated annotation labels. By integrating protein databases with scientific literature into a protein-centric knowledge graph and applying hierarchical graph sampling, ProtSynGen synthesizes QA pairs across atomic, aggregated, and complex reasoning scenarios with diverse instruction forms and evidence-based responses.

Experiments show that ProtSynGen-SFT achieves substantially higher instruction uniqueness, broader semantic coverage, and stronger quality metrics than existing protein LLM instruction datasets. Models trained on ProtSynGen data consistently outperform baselines on open-ended protein QA benchmarks, confirming that the quality of synthetic instruction data directly impacts downstream task performance.

Limitations

This work focuses on open-ended protein question answering and instruction-tuning, where data synthesis and parts of the evaluation protocol involve LLM-based procedures, including the LLM-judge setup described in Appendix A.13 and Ap-

pendix A.14. Such protocols may introduce subtle errors that are inherently difficult to fully detect through automated means. We have conducted manual expert verification to validate the quality of our pipeline (Appendix A.4), and we look forward to future efforts that enable more extensive human evaluation to further strengthen the reliability of synthetic instruction data and benchmark scoring.

Ethics Statement

ProtSynGen is intended for scientific data synthesis, instruction-tuning, and evaluation in the protein domain. Although the pipeline grounds QA generation in curated protein databases and supporting literature, the generated QA pairs and model outputs may still contain incomplete, outdated, or incorrectly inferred biological claims. Therefore, ProtSynGen outputs should not be treated as validated biomedical advice, clinical evidence, or experimental conclusions without expert review.

To reduce factual and safety risks, ProtSynGen includes evidence-grounded generation, quality auditing, homology filtering for evaluation, and manual expert verification on sampled KG edges and QA pairs. These safeguards improve reliability but do not eliminate residual errors, especially for subtle mechanism attribution or biological extrapolation. We encourage users to use the released data and models for research purposes only and to perform task-specific validation before applying them to downstream biological discovery or decision-making.

The resources used in this work are derived from public protein databases and scientific literature. We do not use private patient records or personally identifiable biomedical data.

References

Eli M Carrami and Sahand Sharifzadeh. 2024. Pqa: Zero-shot protein question answering for free-form scientific enquiry with large language models. *Preprint*, arXiv:2402.13653.

Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing Dong. 2025. Graphgen: Enhancing supervised fine-tuning for llms with knowledge-driven synthetic data generation. *Preprint*, arXiv:2505.20416.

Cheng-Han Chiang and Hung-yi Lee. 2023. A closer look into automatic evaluation using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2024. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *ICLR*. OpenReview.net.

Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, 36(2):193–202.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Han Guo, Mingjia Huo, Ruiyi Zhang, and Pengtao Xie. 2023. ProteinChat: Towards achieving ChatGPT-like functionalities on protein 3d structures. TechRxiv preprint.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smestad, Allan Costa, Maryam Salvatore, Abraham Naber, et al. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Zhiyuan Liu, An Zhang, Hao Fei, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. 2024. ProtT3: Protein-to-text generation for text-based protein understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5949–5966, Bangkok, Thailand. Association for Computational Linguistics.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Massimo Hong, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2026. BioMedGPT: An open multimodal large language model for BioMedicine. *IEEE Journal of Biomedical and Health Informatics*, 30(2):981–992.

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. 2024. Prollama: A protein large language model for multi-task protein language processing. *Preprint*, arXiv:2402.16445.

Ying Lv, Yifan Xu, Gang Xu, and Jianpeng Ma. 2026. Opus-pllm: Advancing generative large language models toward discriminative performance in protein function prediction. *Genome Biology*.

Philip M. McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

James W. Pennebaker, Ryan L. Boyd, Kayla Jordan, and Kate Blackburn. 2015. The development and psychometric properties of liwc2015. Technical report, University of Texas at Austin.

Owen Queen, Yepeng Huang, Robert Calef, Valentina Giunchiglia, Tianlong Chen, George Dasoulas, LeAnn Tai, Gianmarco Abbadessa, Owain Howell, Michelle M. Li, Yasha Ektefaie, Ayush Noori, Ildiko Farkas, Joseph Brown, Tom Cobley, Karin Hrovatin, Tom Hartvigsen, Fabian J. Theis, Bradley L. Pentelute, James Zou, Vikram Khurana, David Owen, Richard Nicholas, Manolis Kellis, and Marinka Zitnik. 2024. Procyon: A multimodal foundation model for protein phenotypes. *bioRxiv*.

Xiaofeng Shi, Lulu Zhao, Hua Zhou, and Donglin Hao. 2024. Industrycorpus2.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *Preprint*, arXiv:2405.01724.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. Judgebench: A benchmark for evaluating llm-based judges. *Preprint*, arXiv:2410.12784.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2023. Graphgpt: Graph instruction tuning for large language models. *Preprint*, arXiv:2310.13023.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *Preprint*, arXiv:2404.18796.

Chao Wang, Hehe Fan, Ruijie Quan, and Yi Yang. 2025a. Protchatgpt: Towards understanding proteins with large language models. *Preprint*, arXiv:2402.09649.

Zeyuan Wang, Qiang Zhang, Keyan Ding, Ming Qin, Xiang Zhuang, Xiaotong Li, and Huajun Chen. 2024. InstructProtein: Aligning human and protein language via knowledge instruction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1114–1136, Bangkok, Thailand. Association for Computational Linguistics.

Zhicong Wang, Zicheng Ma, Ziqiang Cao, Changlong Zhou, Jun Zhang, and Yiqin Gao. 2025b. Prot2chat: Protein large language model with early fusion of text, sequence, and structure. *Bioinformatics*, 41(8):btaf396.

Wei Wu, Chao Wang, Liyi Chen, Mingze Yin, Yiheng Zhu, Kun Fu, Jieping Ye, Hui Xiong, and Zheng Wang. 2024. Structure-enhanced protein instruction tuning: Towards general-purpose protein understanding with llms. *Preprint*, arXiv:2410.03553.

Hongwang Xiao, Wenjun Lin, Hui Wang, Zheng Liu, and Qiwei Ye. 2024. OPI: An open instruction dataset for adapting large language models to protein-related tasks. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, and Bowen Yu et al. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xibin Zhou, Chenchen Han, Yingqi Zhang, Huan Du, Jiayuan Tian, Jin Su, Renju Liu, Kai Zhuang, Shiyu Jiang, Anthony Gitter, et al. 2025. Decoding the molecular language of proteins with evolla. *bioRxiv*.

A Appendix

A.1 Full Dataset Statistics

Table 6 reports detailed corpus statistics for the protein LLM instruction-tuning datasets compared in Section 3.1.

A.2 Related Work

A.2.1 Overview

Protein LLMs and protein instruction-tuning. Recent work has increasingly connected protein representations with the instruction-following ability of LLMs. Protein LLMs such as ProtT3 (Liu et al., 2024), BioMedGPT (Luo et al., 2026), InstructProtein (Wang et al., 2024), ProLLaMA (Lv et al., 2024), SEPIT (Wu et al., 2024), ProCyon (Queen et al., 2024), and Prot2Chat (Wang et al., 2025b) encode protein sequences, structures, or multimodal protein features and align them with natural-language LLM interfaces for protein-centric prediction, annotation, question answering, and generation. These systems demonstrate the value of instruction-following interfaces for protein understanding. However, most of them focus on model architecture, modality alignment, or task-specific instruction-tuning, while the construction of diverse and evidence-grounded open-ended instruction data remains less explored. ProtSynGen is complementary to these protein LLMs: rather than proposing a new protein encoder or multimodal architecture, we investigate how to synthesize higher-quality instruction-tuning data across diverse open-ended scenarios.

Instruction data for protein LLMs. Most existing protein LLM instruction datasets are derived from resources such as datasets for protein representation tasks or certain fields in PDB or SwissProt database, converting these protein annotations into protein–question–answer triples. Template-based datasets such as UniProtQA, OPI (Xiao et al., 2024), Mol-Instruction (Fang et al., 2024), and datasets released with ProteinChat (Guo et al., 2023), including SwissProt-V3, ProteinKG25, and PDBQA, instantiate predefined questions over annotation fields, while label-guided generation methods such as Pika-DS (Carrami and Sharifzadeh, 2024) use LLM prompting over structured labels to produce more varied QA pairs. Recent protein LLM efforts further expand the scale, modality coverage, or task scope of protein instruction data. Nevertheless, these datasets are still often tied to predefined task inventories, annotation schemas, or modality-specific supervision. As a result, they may contain repetitive instruction forms or short annotation-style answers. ProtSynGen differs by constructing a protein-centric knowledge graph that links structured annotations with literature-derived evidence, then sampling this graph to generate open-ended QAs at different reasoning depths.

Knowledge Graph-guided data synthesis. Our work is also related to methods that use knowledge graphs to improve question answering or instruction-tuning. UniK-QA (Oguz et al., 2022) studies how structured and unstructured knowledge sources can be unified for open-domain QA, motivating the integration of database facts and textual evidence. GraphGPT (Tang et al., 2023) aligns LLMs with graph structural knowledge through graph instruction-tuning, showing that graph-structured inputs can provide useful supervision for LLMs. GraphGen (Chen et al., 2025) is closely related to our work, as it also uses knowledge-graph based synthetic data generation to improve supervised fine-tuning in general LLMs. ProtSynGen differs by focusing on open-ended protein QA, where supervision must be grounded in protein-specific evidence rather than general knowledge. To this end, we design and construct a protein-centric knowledge graph from databases and scientific literature, and use hierarchical graph sampling to synthesize QAs across atomic, aggregated, and complex reasoning scenarios. Our evaluation also targets protein-specific data issues, including instruction homogenization, reasoning superficiality, temporal protein QA generalization, and expert-scored external questions.

A.3 ProtSynGen Pipeline Implementation Details

Resources and model setup. ProtSynGen uses SwissProt entries as structured protein evidence, supplements them with additional data from BRENDA and Gene Ontology databases, and further enriches them with UniProt expert comments and PubMed abstracts linked from SwissProt references. For the final production pipeline, we use GLM-4-Air for graph enrichment and QA synthesis, and DeepSeek-v3 for quality audit. We set a low temperature (0.1) for extraction/enrichment and audit, and a higher temperature (0.7) for QA generation. The low-temperature setting is intended to improve stability and factual precision in graph construction and auditing, while the higher

Dataset	#Samples	Template	Unique Count				Average Length		
			Instruction	Response	Sequence	Inst. Ratio (%)	Inst. Tokens	Resp. Tokens	Sequence
SwissProt-V3	430595	✓	–	98019	430595	–	–	75.80	335.88
ProteinKG25	422315	✓	–	314540	422315	–	–	141.45	337.69
Mol-Instruction	299029	✓	60	97004	298925	0.020	24.24	59.32	397.81
OPI	1776636	✓	161	305921	485133	0.009	19.01	31.87	372.68
UniProtQA	1513126	✓	4	247825	391836	0.000	11.30	25.85	370.27
PDBQA	3359693	✓	30	82356	51298	0.001	12.46	3.90	291.45
Pika-DS	1714885	✗	97255	782907	251532	5.671	10.21	17.81	412.50
ProtSynGen-SFT	5162352	✗	4141715	5024733	483954	80.229	29.71	179.11	371.60
ProtSynGen-SFT by reasoning scenario									
Atomic	3901433	✗	2881647	3763815	481739	73.861	23.58	85.05	372.43
Aggregated	1242882	✗	1242034	1242881	441716	99.932	48.78	467.63	368.90
Complex	18037	✗	18037	18037	17650	100.000	40.56	642.79	378.91

Table 6: Detailed statistics of protein LLM instruction-tuning datasets. Inst. Ratio denotes the percentage of distinct instructions. Token lengths are computed with SapBERT-PubMedBERT, following Table 1. SwissProt-V3, ProteinKG25, and PDBQA are from ProteinChat (Guo et al., 2023).

temperature encourages more diverse open-ended responses.

Base KG schema and enrichment constraints. The base protein-centric graph is initialized from seven core field types: protein, gene ontology, ec_numbers, keywords, organism, host_organism, and gene_name. Literature enrichment is constrained by (i) the current protein-centered graph and (ii) a fixed biological entity-type inventory, so new entities and relations remain connected to protein evidence rather than drifting into unconstrained free-text generation. The entity inventory includes core biomolecular entities (e.g., Protein, Gene, Protein Complex, Molecule, Nucleic Acid), functional entities (e.g., Biological Process, Molecular Function, Cellular Component, Pathway, EC Numbers, Subcellular Localization), context entities (e.g., Organism, Host organism, Tissue/Organ, Cell Type), and phenotype/regulation entities (e.g., Disease/Phenotype, Regulator, Interaction Partner). Relation candidates are further tagged as direct document support or reasoning-based inference.

Entity normalization and disambiguation. We further apply a lightweight entity normalization module during graph insertion and merge. Candidate entity names are embedded with SapBERT and matched by cosine similarity under a high threshold, with small LLM-based (GLM4-Flash) fuzzy matching used only for unresolved ambiguous cases. This stage reduces duplicate aliases and near-duplicate entities before graph expansion and downstream sampling.

Literature scale and graph-conditioned sampling. In our SFT literature pool, the collected abstract chunks contain 266,308 unique PMIDs. For the 10K-protein training subset, 8,270 proteins have at least one available PubMed abstract, with 2.07 available abstracts per protein on average. During sampling, atomic and aggregated candidates are proposed with weights 0.75/0.25, with at most 15 non-complex sampled batches per protein. Complex reasoning generation is enabled only when the diameter of the local protein-centered graph exceeds a predefined threshold, making reasoning-level assignment graph-conditioned rather than prompt-only.

Quality audit behavior. The audit stage is implemented as a conservative binary filter. A QA pair is rejected if it fails any of the following checks: protein-centricity/anonymization, answer validity, factual support from raw evidence, or consistency with sampled graph evidence. This audit is used for filtering, not as a substitute for expert biological validation; manual expert verification is reported in Appendix A.4. Representative prompt assets across extraction, enrichment, sampling, and audit are provided in Appendix B.

A.4 Intrinsic Quality Validation of KG and QA Generation

To assess the factual reliability of the ProtSynGen pipeline independently of downstream task performance, we conduct expert-level verification on a random sample of 15 proteins from the enriched knowledge graph. The audit covers two levels: (1) all 979 edges present in these proteins' knowledge subgraphs, and (2) 100 QA pairs generated from these proteins and processed through the quality audit stage (Section 2.4).

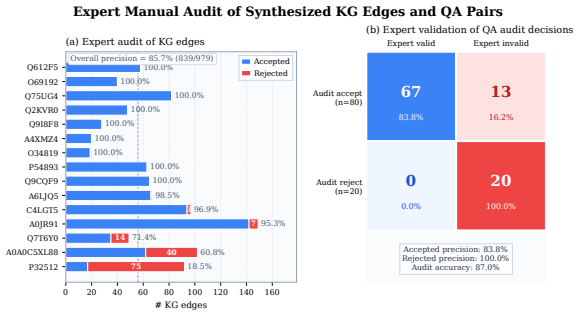

Figure 6: Manual expert audit of synthesized KG edges and QA pairs. (a) Per-protein edge audit results sorted by precision; the dashed line marks the mean accepted count. (b) Confusion matrix of LLM quality audit decisions vs. expert verification on 100 QA pairs.

Audit Setup. The 15 proteins are drawn to cover diverse organisms (bacteria, archaea, viruses, mammals) and functional classes (enzymes, toxins, structural proteins, oxidoreductases). Their UniProt accessions are: O34819, O69192, A0A0C5XL88, A0JR91, A4XMZ4, A6LJQ5, C4LGT5, P32512, P54893, Q2KVR0, Q612F5, Q75UG4, Q7T6Y0, Q9CQF9, Q9I8F8. Each edge and QA pair is evaluated against the source SwissProt entry and all cited PubMed abstracts on three criteria: **Factivity** (is the claim supported by source evidence?), **Validity** (does the answer correctly address the protein-centric question?), and **Self-Consistency** (is the QA consistent with the sampled subgraph or reasoning path?).

Edge-Level Factual Precision. Of the 979 edges audited, 839 (85.7%) are confirmed factually correct by expert review (Figure 6a). Specifically, the 140 rejected edges fall into three failure modes:

- **Concept inflation** (89/140): The LLM generates hierarchies of related concepts from a functional keyword without grounding in the specific protein's evidence. For example, a baculovirus transcription factor named "Major Immediate Early Protein" triggers fabricated edges about viral replication machinery.
- **Function class confusion** (34/140): The LLM attributes functions typical of a well-known protein class to a different member. For example, a bacterial toxin–antitoxin component named "ADP-ribosylation toxin" is treated as a classical pathogenic virulence factor targeting host cells, when it actually only targets its own cell's ribose-phosphate pyrophosphokinase.
- **Wrong attribution** (17/140): Incorrect EC numbers, reversed reaction directions, or fabricated post-translational modifications assigned without evidence.

Notably, 92.1% of errors (129/140) originate from just 3 of the 15 proteins—all cases where the protein name is lexically similar to a better-known functional class. The remaining 12 proteins achieve 98.2% edge-level precision, indicating that enrichment errors are not uniformly distributed but cluster around specific naming ambiguities.

QA-Level Audit Validation. Figure 6(b) reports the confusion matrix between the pipeline's LLM quality audit decisions and expert verification on 100 QA pairs.

The audit exhibits two desirable properties in this sampled audit set. First, **zero false rejections**: all 20 items filtered by the audit are confirmed problematic, indicating that the audit is conservative in the examined cases. Second, a **post-audit acceptance precision of 83.8%** (67/80), confirming that the large majority of retained QA pairs are factually grounded. The overall accuracy is 87%.

Case Analysis. We present representative cases to illustrate the audit's behavior and the nature of residual errors.

Case 1: Correct rejection (True Positive). QA-024 asks "What role does pectin lyase play in the integration of prophage genetic material into the host's genome?" The generated answer fabricates a connection between pectin degradation and prophage integration. The audit correctly identifies this as unsupported by any cited evidence and rejects it.

Case 2: Subtle error accepted by audit (False Negative). QA-014 asks "How does this protein affect the host's memory response to pathogens?" for ParT (A0A0C5XL88), a Type II toxin–antitoxin system component from *Sphingobium* sp. The answer describes host immune memory disruption—a function characteristic of classical pathogenic ADP-ribosylation toxins (e.g., pertussis toxin) but not of intracellular TA system toxins that only target their own cell's RPPK. The audit accepts this because the answer uses correct protein terminology and produces biologically fluent text, but the underlying mechanism is misattributed from a different protein class.

Case 3: Edge-level concept inflation. For P32512 (a baculovirus transcription factor), the

enrichment LLM generates edges such as "viral replication adaptation → viral replication evolution." Neither concept appears in the SwissProt entry or cited PubMed abstracts. The LLM inferred these from the protein's classification as an "immediate early protein," over-generalizing to viral replication concepts without specific evidence.

Summary. The quality audit implements a conservative filtering strategy that prioritizes preserving valid training data (zero false rejections) while removing clearly problematic items. The residual errors that pass the audit are predominantly subtle mechanism misattributions that require deep domain expertise to detect. These errors are concentrated in a small number of proteins with ambiguous naming, and their impact on downstream SFT is likely mitigated by dilution across the 5.16M QA pairs. Future work may incorporate domain-specific verification modules (e.g., biological agent validation) to further reduce these residual errors.

A.5 Case Study

Figure 7 shows an aggregated case that asks how a protein contributes to the producing organism's defense strategy. Among the compared models, only the model trained on ProtSynGen-SFT accurately identifies the protein as neurotoxin and explains its neurotoxin-mediated predator-deterrence function, supported by SwissProt annotation and PubMed evidence.

A.6 Instruction Diversity and Topic Labeling

For the UMAP analysis in Figure 3, we use all explicit instructions from template-based datasets because their instruction pools are small. For Pika-DS and ProtSynGen-SFT, we use matched 10K instruction samples. Instructions are encoded with SapBERT-PubMedBERT and reduced to two dimensions with UMAP using cosine distance, $n_{\text{neighbors}} = 30$, $\text{min_dist} = 0.08$, and random seed 42.

For the topic-composition analysis in Figure 4, we cluster the matched 10K instruction samples from Pika-DS and ProtSynGen-SFT into 50 semantic topics and assign a short topic label to each cluster with an LLM based on 10 representative instructions from each cluster. Figure 8 reports the full topic-proportion matrix. Each cell denotes the percentage of instructions assigned to a topic within the corresponding 10K sample. Blank cells indicate that no sampled instruction is assigned to

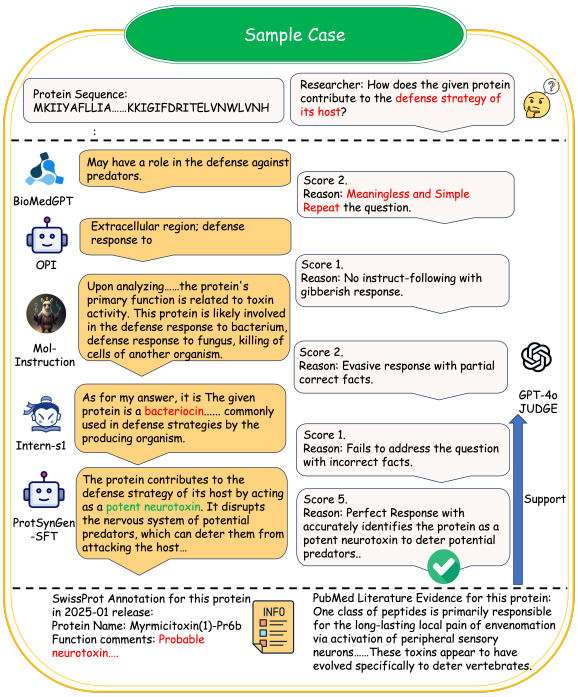

Figure 7: An aggregated sample case. Only ProtSynGen-SFT accurately identifies the protein's neurotoxin-mediated predator-deterrence function using SwissProt annotation and PubMed evidence.

that topic, while $\leq 0.1\%$ indicates a nonzero but very small topic share.

A.7 Term-Based Reasoning Depth Indicators

Figure 5 uses lightweight lexical indicators to estimate response-level reasoning depth. This analysis is motivated by prior text-analysis work such as Coh-Metrix and LIWC, which use discourse, causal, cognitive, and lexical categories to characterize cohesion, explanatory relations, and cognitive content in text (Graesser et al., 2004; Pennebaker et al., 2015). We adapt this idea to protein QA data by counting two types of terms in model responses: **reasoning terms** and **biological mechanism terms**. These indicators are used as proxy diagnostics rather than as direct measures of factivity.

For each dataset, we sample up to 10K responses using a fixed random seed. For ProtSynGen-SFT, we sample 10K responses from each reasoning scenario: atomic, aggregated, and complex reasoning. Let a_i be the answer text of instance i, and let N_i be its token count computed with a regex tokenizer. For a term set $\mathcal{M}$, we compute a normalized den-

sity:

$$\text{Density}_{\mathcal{M}} = \frac{\sum_i \text{count}_{\mathcal{M}}(a_i)}{\sum_i N_i} \times 100. \quad (12)$$

We report densities per 100 response tokens and estimate confidence intervals using bootstrap resampling over sampled responses.

Reasoning terms include causal, contrastive, evidential, inferential, and conditional expressions, such as *because*, *due to*, *therefore*, *as a result*, *whereas*, *however*, *suggests*, *indicates*, *implies*, *based on*, *consistent with*, *likely*, *may*, and *could*. These terms approximate whether a response explicitly links statements through reasoning relations.

Biological mechanism terms include biology-specific action and relation expressions, such as *binding*, *interaction*, *regulation*, *activation*, *inhibition*, *mediation*, *catalysis*, *transport*, *localization*, *phosphorylation*, *pathway*, *signal transduction*, *substrate*, *cofactor*, *active site*, *complex formation*, *stabilization*, and *degradation*. This axis is included as a linguistic estimate: it helps distinguish generic reasoning language from explanations that remain grounded in protein-related biological mechanisms.

Because these indicators are lexical proxies, they should not be interpreted as complete measures of biological reasoning. In particular, a high density of biological mechanism terms can arise from short annotation-style answers that repeatedly use action-oriented labels. We therefore interpret the two axes jointly: reasoning terms indicate explicit explanatory links, while biological mechanism terms indicate whether those links are accompanied by protein mechanism-related content.

A.8 Data-Quality Metrics

Table 2 reports three complementary metrics used to assess whether gains in diversity and response depth come at the cost of data quality.

MTLD (McCarthy and Jarvis, 2010) measures lexical diversity while being relatively stable to text length compared with simple type-token ratios. We use it as a corpus-level indicator of whether the generated responses exhibit varied surface realization rather than repetitive lexical reuse.

UniEval (Zhong et al., 2022) provides model-based quality scores for **naturalness**, **coherence**, and **understandability**. In this section, these scores are used as lightweight proxies for whether broader and longer responses remain readable and instruction-following.

Reward_Ind (Shi et al., 2024) is a reward-model score designed for instruction-tuning data quality evaluation. We use it as an additional preference-style signal complementary to lexical diversity and surface text quality.

These metrics complement the diversity and reasoning-depth analyses by testing whether broader semantic coverage and longer responses are accompanied by acceptable linguistic quality.

A.9 Model Architecture and Training Hyperparameters

Model Architecture. Our protein LLM follows a modular multimodal design consisting of three components: (1) a protein sequence encoder based on ESM-2 (650M parameters, 33-layer) (Lin et al., 2023), which produces per-residue representations that are mean-pooled into a fixed-dimensional sequence embedding (1280-d); (2) an MLP projector that maps the protein embedding into the LLM's text embedding space (5120-d); and (3) Qwen2.5-7B-Instruct (Yang et al., 2025) as the language backbone. The protein embedding is inserted as a special <seq> token in the input sequence, replacing the token embedding at that position.

Training Pipeline. Training proceeds in two stages. **Stage 1 (Modality Alignment):** The MLP projector is trained to align protein sequence representations with the LLM's text space, using protein caption data from OPI (Xiao et al., 2024). During this stage, only the MLP projector is trainable; the ESM-2 encoder and LLM backbone are frozen. All models in the retraining comparison share this identical alignment stage. **Stage 2 (instruction-tuning):** Starting from the shared alignment checkpoint, each dataset is used to fine-tune both the MLP projector and the full LLM backbone. The ESM-2 encoder remains frozen throughout. Table 7 summarizes the Stage 2 hyperparameters.

A.10 Training Corpus Size

To compare retraining corpora by token budget, we tokenize each SFT dataset with the Qwen2.5-7B-Instruct tokenizer used by the language backbone. Table 8 shows that ProtSynGen-SFT-106K is the smallest retraining corpus in both number of records and total token count.

Hyperparameter	Value
Base model	Qwen2.5-7B-Instruct
Protein encoder	ESM-2 (650M, frozen)
Trainable parameters	MLP projector + LLM backbone
Optimizer	AdamW
Learning rate	1e-5
LR scheduler	Cosine
Warmup ratio	0.01
Weight decay	0.0
Epochs	1
Per-device batch size	12
Number of GPUs	6 × H200
Max sequence length	1024
Precision	bf16 (DeepSpeed ZeRO-2)

Table 7: Stage 2 instruction-tuning hyperparameters.

Dataset	#Records	#Tokens
OPI	1,768,252	460,867,066
PDBQA	3,359,693	588,831,756
UniProtQA	1,513,126	368,875,356
Pika-DS	1,714,885	438,194,762
Mol-Instruction	299,029	92,268,979
ProtSynGen-SFT-106K	106,818	45,436,641

Table 8: Training corpus size under the Qwen2.5-7B-Instruct tokenizer.

A.11 Supplementary Reference-Based Metrics on LiveProtQA

In addition to judge-based evaluation, we report conventional reference-based text generation metrics on LiveProtQA as supplementary lexical-semantic signals. Following the same model grouping used in Table 3, we report BLEU, ROUGE-2, ROUGE-L, BERTScore (BioMedBERT and PubMedBERT), and METEOR on a 0–100 scale. However, the language styles and open-ended scenarios significantly influence these reference-based metrics, so they are not treated as primary factuality criteria.

A.12 Benchmark Construction and Homology Filtering

LiveProtQA. LiveProtQA is constructed from four held-out SwissProt snapshots: 27-NOV-2024 (SwissProt 2024_06), 05-FEB-2025 (SwissProt 2025_01), 09-APR-2025 (SwissProt 2025_02), and 18-JUN-2025 (SwissProt 2025_03). Using these held-out snapshots as source evidence, we build the benchmark to cover atomic, aggregated, and complex reasoning scenarios, followed by deduplication, light balancing, and sequence-level homology filtering. The final benchmark contains 1,800 questions covering 935 protein sequences, including 760 atomic, 939 aggregated, and 101 complex-reasoning items.

ExpertProtQA. To provide an externally sourced evaluation, we curate ExpertProtQA from existing protein-biology benchmarks that can be adapted into open-ended protein QA. Most prior benchmarks in this area emphasize classification, tagging, captioning, or multiple-choice evaluation rather than free-form answers grounded in a target protein sequence. We therefore select and normalize suitable items from Pika-DS, OpusPLLM (Lv et al., 2026), and QAProt, yielding 118 open-ended questions covering 99 proteins. ExpertProtQA is evaluated by two invited domain experts and by LLM judges.

Homology filtering. For both LiveProtQA and ExpertProtQA, we first export unique benchmark protein sequences as MMseqs2 queries and unique ProtSynGen-SFT-106K training sequences as the target database, then run MMseqs2 easy-search to compute the best training-set hit for each benchmark sequence. A benchmark item is removed if its protein sequence has at least 80% identity to any training sequence; otherwise, the item is retained.

A.13 Evaluation Details

Pointwise scoring protocol. Each response is judged independently on a 1–5 scale. The rubric emphasizes two criteria: **helpfulness**, i.e., whether the response addresses the question and provides useful biological information, and **factivity**, i.e., whether the response remains consistent with the supported protein evidence. Limited reasonable biological inferences are allowed as long as they remain scientifically plausible.

LLM-judge setup. Both LiveProtQA and ExpertProtQA are scored by three independent LLM judges (GPT-4o, DeepSeek-R1, Qwen3-235B). Each judge follows the same pointwise rubric and assigns a 1–5 score; we report the mean of the three judges and linearly normalize it to 0–100. This design follows the broader LLM-as-a-judge literature, which shows that strong judges can approximate human preferences in open-ended evaluation while also exhibiting prompt sensitivity, position bias, verbosity bias, and other failure modes that should be checked explicitly (Zheng et al., 2023; Liu et al., 2023; Chiang and Lee, 2023; Stureborg et al., 2024; Verga et al., 2024; Tan et al., 2024; Gu et al., 2025).

Human evaluation on ExpertProtQA. ExpertProtQA is also scored pointwise by domain experts

Model Source	Model/Dataset	BLEU ↑	ROUGE ↑		BERTScore ↑		METEOR ↑
			R-2	R-L	BioMed	PubMed	
Publicly released checkpoints							
Public Release	InstructProtein	0.00	0.04	0.97	88.49	38.51	0.74
	OPI-Galactica	0.00	0.79	3.62	88.39	49.06	2.02
	BioMedGPT	0.00	3.08	7.62	90.51	56.60	4.12
	Intern-S1	1.30	10.57	16.94	92.53	74.30	13.30
Retrained with the same architecture and different datasets							
Retrain	Qwen2.5-7B-Instruct	7.32	11.82	16.90	92.37	76.55	7.18
	PDBQA	0.00	0.01	0.30	86.65	38.52	0.10
	OPI	0.08	2.92	9.22	90.43	63.06	6.43
	UniProtQA	0.30	4.29	10.94	90.70	66.00	7.85
	Mol-Instruction	0.37	6.60	13.48	91.69	71.05	9.79
	Pika-DS	0.01	11.52	16.45	93.20	76.23	9.73
	ProtSynGen-SFT-106K	**20.93**	**24.30**	**30.70**	**94.47**	**84.33**	**41.54**

Table 9: Reference-based text generation metrics on LiveProtQA. All metrics are reported on a 0–100 scale. BioMed and PubMed denote BERTScore F1 computed with BioMedBERT and PubMedBERT/SapBERT encoders, respectively. These metrics are supplementary lexical/semantic matching signals rather than primary factuality judgments.

using the same helpfulness/factivity emphasis. We invited two biological researchers to independently score all question-answer pairs, with model outputs randomized to prevent ordering bias. We use expert scores as the primary metric and additionally report LLM-judge scores as a secondary reference.

A.14 LLM-as-a-Judge Reliability Analysis

Because our main benchmarks rely partly on LLM-based scoring, we further audit the judge protocol itself. Prior work has shown that LLM judges can align well with human preferences on open-ended tasks, but they may also be affected by verbosity, position, anchoring, and prompt-format choices (Zheng et al., 2023; Liu et al., 2023; Chiang and Lee, 2023; Stureborg et al., 2024; Verga et al., 2024; Tan et al., 2024; Gu et al., 2025). We therefore use multiple complementary checks rather than relying on a single judge outcome.

On ExpertProtQA, we anchor the LLM-judge scores to human ratings and report agreement diagnostics in Figure 9. Across ExpertProtQA, LLM-judge scores preserve the same model ranking as expert scores and show strong agreement with expert evaluation (within-1 agreement 95.9%; pairwise direction agreement 89.4%). Let $h_{q,s}$ denote the expert score for system s on question q, and let $\hat{y}_{q,s}$ denote the mean score of the three LLM judges. Exact agreement is the proportion of (q, s) pairs for which $\mathrm{round}(\hat{y}_{q,s}) = h_{q,s}$, while within-1 agreement is the proportion for which $|\mathrm{round}(\hat{y}_{q,s}) - h_{q,s}| \leq 1$. Pairwise direction agreement is computed over all question-specific system pairs (s_i, s_j) with $h_{q,s_i} \neq h_{q,s_j}$, and measures the proportion for which $\hat{y}_{q,s_i} - \hat{y}_{q,s_j}$ has the same sign as $h_{q,s_i} - h_{q,s_j}$. These results suggest that the judge protocol is sufficiently aligned with the human reference to serve as a secondary metric.

On LiveProtQA, we score each answer with three judges and report the mean score as the primary aggregate. Figure 10 shows that model rankings remain stable across judges, while the mean and median aggregates remain closely correlated with the individual judges. This supports the use of the mean judge score as a stable evaluation signal rather than relying on any single judge's preference.

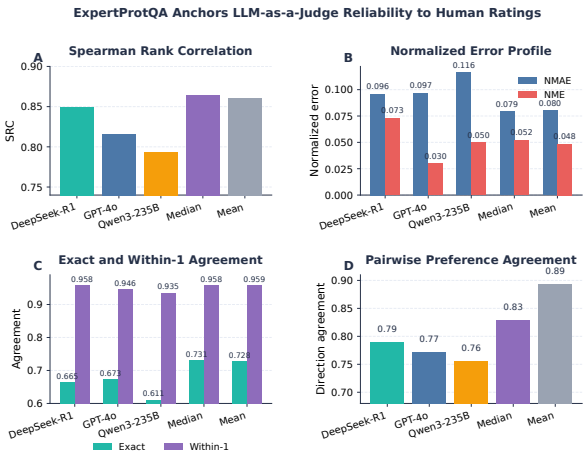

Figure 9: ExpertProtQA judge reliability anchored to human ratings. The LLM judges preserve the human ranking and achieve high agreement with expert scores.

Figure 11 examines whether judge scores are driven by superficial response features. We find that judge sensitivity to answer length and question length is limited after aggregation, while score patterns across reasoning types and topics remain interpretable rather than erratic. This analysis is not meant to claim complete judge invariance, but to show that the reported main conclusions are not explained by a single obvious surface bias.

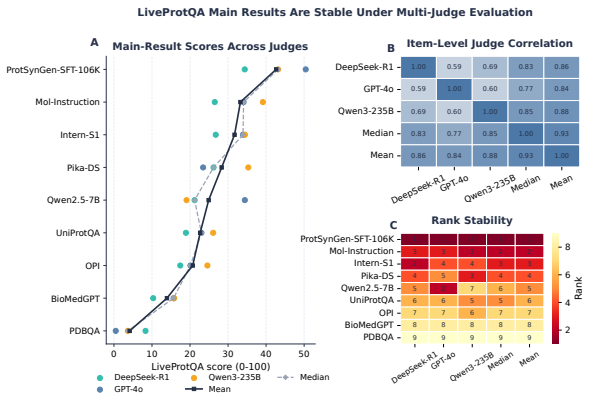

Figure 10: LiveProtQA scores are stable across judges. We report the mean judge score as the primary aggregate and use the median as a robustness reference.

A.15 Ablation Significance Tests

Statistical tests. For LiveProtQA ablations, we compare per-item paired scores against the full ProtSynGen setting using two-sided paired Wilcoxon signed-rank tests ($n = 1{,}800$). For ExpertProtQA, we apply the same protocol on expert scores ($n = 118$). We also report bootstrap 95% confidence intervals on mean score differences using paired resampling.

Results summary. Removing KG enrichment or quality audit yields consistent negative deltas on both benchmarks. On LiveProtQA, both ablations are significant at $p < 0.001$ (paired Wilcoxon), with bootstrap 95% confidence intervals excluding zero. On ExpertProtQA, both ablations reduce expert scores by 5.08 points and are significant at $p < 0.05$, again with confidence intervals excluding zero. Table 10 reports the full ExpertProtQA ablation results.

Win–loss–tie pattern. Item-level comparison shows that the full model wins more often than each ablated variant on all three LiveProtQA scenarios (atomic, aggregated, complex), with the largest win margin on complex reasoning.

Variant	Expert Score ↑
Full	**48.09**
w/o KG Enrichment	43.01
w/o Quality Audit	43.01

Table 10: Ablation results on ExpertProtQA. Scores are normalized to 0–100.

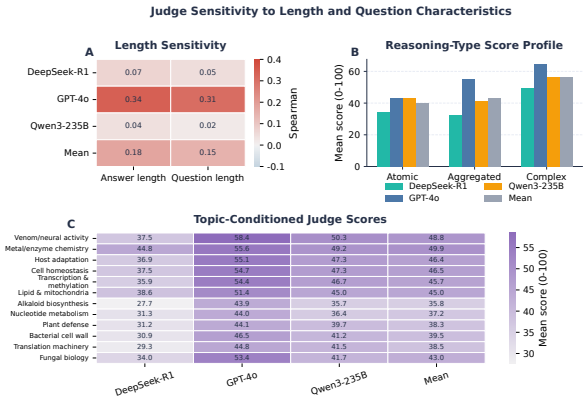

Figure 11: Judge sensitivity to answer length, question length, reasoning type, and topic. Mean aggregation reduces the apparent surface-feature dependence relative to any single judge.

Figure 12: Ablation consistency on LiveProtQA. The same conclusion holds across judges, reasoning scenarios, and aggregate statistics, indicating that the ablation result is not driven by a single judge.

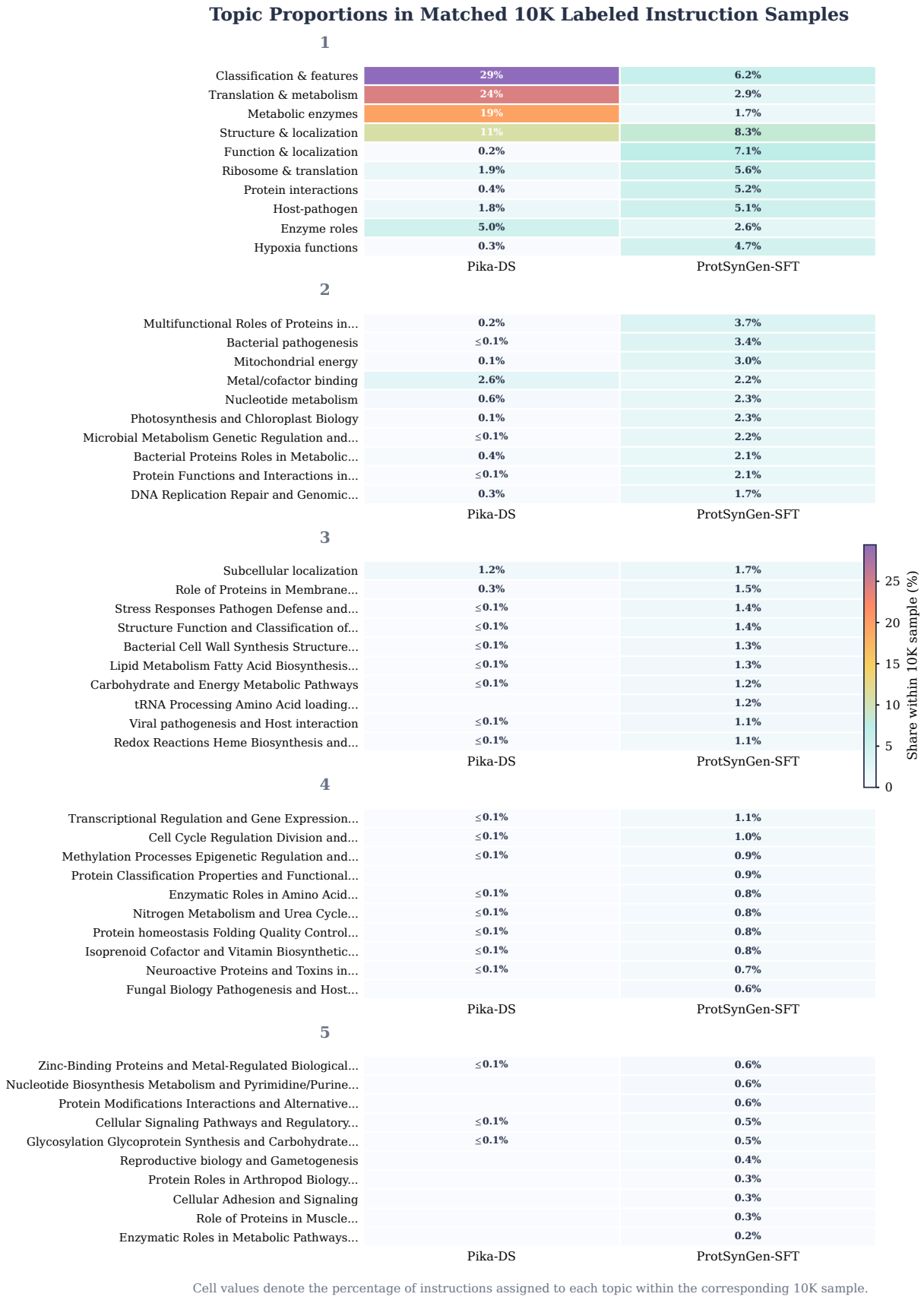

Figure 8: Full 50-topic proportions for matched 10K instruction samples from Pika-DS and ProtSynGen-SFT. Cell values denote topic shares within each dataset sample.

B Prompt Library

This appendix visualizes the main prompt templates used in the implemented ProtSynGen pipeline. Long few-shot demonstrations are replaced by Case placeholders for readability.

Entity Discovery Prompt

```
Assume you are both a biological research expert and an NLP expert. Your task is to read a
protein-centered basic knowledge graph and the literature context associated with the same
protein, then extract new biological entities that can enrich the graph.

Goal:
The basic KG is constructed from structured UniProt annotations and contains core entities
around the target protein. The literature context contains UniProt functional comments and
PubMed abstracts related to the protein. The KG should not be rewritten; it should be used as
grounded context for discovering additional entities. Enrich the basic KG by reading these
documents and extracting entities that can be directly or indirectly linked with the existing
graph.

Inputs:
- Protein name: {protein_name}
- Basic KG entities: {Given_KG}
- Literature context: {Given_Documents}
- Allowed entity type set: {entity_type}

Entity requirements:
1. Identify biological concepts that are directly or indirectly linked with the existing KG
entities.
2. Do not create duplicate entities already present in the KG, even if the literature uses a
slightly different surface form.
3. Ignore publication metadata such as journal names, DOI, author names, affiliations, and
indexing information.
4. Do not extract entities that are unrelated to biological content.
5. Assign each new entity to one type from the provided schema-defined type set.
6. Preserve the basic KG as grounding context; do not rewrite, delete, or override existing
entities.

For each extracted entity, provide:
- entity_name: the entity name, using the language of the source text where appropriate.
- entity_type: one schema-defined biological entity type.
- entity_summary: a concise summary of the entity's biological attributes, activity, or role.

Output:
Return only a delimiter-separated list of entity tuples in the form:
("entity", entity_name, entity_type, entity_summary)
End the output with <|COMPLETE|>.

[Few-shot demonstration in implementation: Case 1. Full biological example omitted for space.]
```

Relation Discovery Prompt (1)

```
Assume you are both a biological research expert and an NLP expert. Given two entity lists and
the literature context, identify biologically meaningful relations that can be inserted into the
enriched protein-centered KG.

Goal:
The Main Entity List contains entities that need to be connected with other graph entities. The
Candidate Entity List contains existing or newly discovered entities that may provide useful
biological context. The literature context is used to decide whether relations are directly
supported by text or only supported by basic biological reasoning. You may also connect existing
KG entities when the document or simple biological knowledge supports the relation.

Inputs:
- Main entity list: {Main_Entity_List}
- Candidate entity list: {Candidate_Entity_List}
- Literature context: {Given_Documents}
```

Relation Discovery Prompt (2)

```
Relation requirements:
1. Select pairs of entities that are clearly related according to the provided literature
context or basic biological reasoning.
2. Entity names in the output must exactly match names from the Main Entity List or Candidate
Entity List.
3. Do not invent a new entity while generating a relation.
4. Use Documents when the relation is directly supported by the literature context.
5. Use Reasoning when the relation is not explicitly stated but follows from biological
reasoning over the provided entities and context.
6. Do not output unrelated relation candidates or extra explanations outside the tuple format.
7. Include biologically plausible relations only when they can be justified from the provided
text or simple domain reasoning.

For each relation, provide:
- source_entity: the source entity name.
- target_entity: the target entity name.
- relationship_summary: a short explanation of the biological connection.
- evidence_level: Documents or Reasoning.

Output:
Return only a delimiter-separated list of relation tuples in the form:
("relationship", source_entity, target_entity, relationship_summary, evidence_level)
End the output with <|COMPLETE|>.

[Few-shot demonstrations in implementation: Case 1 and Case 2. Full biological examples omitted
for space.]
```

Atomic Question Prompt

```
There is one target protein, one related biological entity, and one direct relation between
them. A short evidence summary provides additional biological context. Generate five simple
protein-centric questions focused on this direct relation.

Task setting:
The questioner is assumed to know only the protein sequence and does not know the protein name
or its annotations. The question should therefore ask about the biological concept or property
of the target protein without naming it. The five questions should be paraphrased variants of
the same evidence-backed query rather than unrelated prompts.

Inputs:
- Entities: {entities}
- Direct relation: {relation}
- Evidence summary: {summary}
- Language style: {language_style}

Question requirements:
1. The target protein must remain anonymous in the question.
2. Refer to the protein using generic expressions such as "this protein" or "the given protein".
3. Each question should be simple, concise, and centered on the given relation.
4. The five questions should preserve the same core meaning, while varying interrogative format
or major synonyms.
5. Do not ask about facts that are not supported by the entity-relation evidence.
6. Use the requested language style consistently.

Output:
Return exactly five numbered questions and no additional text.

[Few-shot demonstrations in implementation: Case 1 and Case 2. Full biological examples omitted
for space.]
```

Atomic Answer Prompt

```
Assume you are a biological researcher answering a student's protein-centric question. The
student refers to an anonymous protein and asks about one direct biological relation. Use the
provided entities, relation, and summary as supporting evidence.

Inputs:
- Entities: {entities}
- Direct relation: {relation}
- Evidence summary: {summary}
- Question: {question}

Answer requirements:
1. Address the student's question directly and briefly.
2. Ground the answer in the sampled entity-relation evidence.
3. Do not disclose that you were given hidden evidence, a sampled subgraph, or a generation
process.
4. The answer should sound natural, fluent, and professional, like a concise academic
discussion.
5. Avoid unsupported speculation beyond the provided evidence.
6. Treat the supplied evidence as background knowledge rather than something to quote
explicitly.

Output:
Return only the final answer.
```

Aggregated Question Prompt

```
There is one target protein, several related biological entities, and internal relations among
them. A summary describes how these entities are connected around the target protein. Generate
five aggregated protein-centric questions.

Task setting:
Aggregated questions should go beyond one isolated protein-to-label fact. They should combine
multiple basic facts, add useful biological constraints, or request a structured answer while
remaining answerable from the sampled subgraph and summary. The question should cover as much
relevant information as possible from the provided entities, relations, and summary without
changing the biological facts.

Inputs:
- Entities: {entities}
- Relations: {relations}
- Evidence summary: {summary}

Question requirements:
1. The target protein must remain anonymous in the question.
2. Refer to the protein using generic expressions such as "this protein" or "the given protein".
3. Each question should involve multiple biological concepts or relations from the provided
evidence.
4. Questions may include format constraints, comparative framing, biological settings, or
structured-output requirements.
5. Do not change the biological facts in the sampled evidence.
6. Do not create questions that require unsupported external knowledge.
7. Keep the anonymous-reference requirement for the target protein.

Output:
Return exactly five numbered aggregated questions and no additional text.

[Few-shot demonstrations in implementation: Case 1 and Case 2. Full biological examples omitted
for space.]
```

Aggregated Answer Prompt

```
Assume you are a biological researcher answering an aggregated protein-centric question. The
question may involve multiple biological concepts, relations, or output constraints. Use the
sampled entities, relations, and summary as supporting evidence.

Inputs:
- Entities: {entities}
- Relations: {relation}
- Evidence summary: {summary}
- Question: {question}

Answer requirements:
1. Answer comprehensively while staying grounded in the sampled evidence.
2. If the question requests a specific format or constraint, follow it strictly.
3. Address the biological relationships among the protein and related entities, rather than only
restating a single label.
4. Do not reveal the existence of hidden evidence, the subgraph, or the generation process.
5. Write in a natural, fluent, and professional style suitable for academic discussion.
6. Avoid unsupported claims beyond the provided evidence.
7. Prefer a compact but complete explanation that covers the relevant relations in the sampled
graph.

Output:
Return only the final answer.
```

Complex Question Prompt

```
Assume you are a biological researcher familiar with proteins and related biological concepts.
Your task is to formulate complex reasoning questions around a target protein for realistic
biological reasoning or research scenarios.

Goal:
The provided entities and relations come from the protein-centered KG, and the context contains
functional comments and paper abstracts. Generate questions that require multi-hop reasoning
over graph evidence and literature context, rather than short one-hop annotation lookup. The
questions should resemble realistic research or analysis scenarios where a reader needs to
connect annotations, literature evidence, and the target protein.

Inputs:
- Main protein: {main_protein}
- Entities: {entities}
- Relations: {relations}
- Literature context: {context}

Question requirements:
1. Each question must focus on the main protein.
2. The main protein must be anonymized in the question using expressions such as "this protein"
or "the given protein".
3. Questions should require multi-hop reasoning across entities, relations, and literature-
derived context.
4. Questions should be biologically meaningful and resemble realistic research or analysis
scenarios.
5. Avoid questions that can be answered by copying a single annotation label.
6. Do not introduce unsupported entities or relations outside the provided evidence.
7. Keep the main protein anonymous so the question remains protein-sequence centered.

Output:
Return only numbered complex reasoning questions.

[Few-shot demonstrations in implementation: Case 1 and Case 2. Full biological examples omitted
for space.]
```

Complex Reasoning-Path Prompt

```
Given a complex protein-centric question and the same graph evidence used to generate it,
construct reasoning paths that can support a later answer. Each path should show how relations
in the sampled subgraph can be connected into a biologically meaningful reasoning chain.

Inputs:
- Complex reasoning question: {question}
- Main protein: {main_protein}
- Entities: {entities}
- Relations: {relations}
- Literature context: {context}

Reasoning-path requirements:
1. Each path must contain Relationships, Description, and Implication.
2. Relationship indexes must be consistent with the provided relation list.
3. Every relation used in a path must actually exist in the provided graph evidence.
4. The path should be useful for constructing an answer to the complex question.
5. The path should conform to biological facts in the sampled graph and literature context.
6. Do not invent additional relations, entities, or unsupported biological claims.
7. Prefer paths that are short but complete enough to support a clear multi-hop answer.

Output:
Return only the reasoning paths. Do not add extra commentary after the paths.

[Few-shot demonstrations in implementation: Case 1 and Case 2. Full biological examples omitted
for space.]
```

Complex Answer Prompt

```
Assume you are a biological expert answering a student's complex protein-centric question. You
are given reasoning paths that connect graph relations and literature-derived implications.
Rewrite these paths into a detailed, multi-hop answer.

Inputs:
- Complex reasoning question: {question}
- Reasoning paths: {paths}

Answer requirements:
1. Start by analyzing what the question is asking about the anonymous protein.
2. Explain the answer step by step, using the relationships and implications from the reasoning
paths.
3. Make the reasoning progressive: each step should clarify how one biological fact leads to the
next.
4. Conclude with a comprehensive answer that directly addresses the question.
5. Do not reveal the existence of reasoning paths or the generation process.
6. Use a natural, professional, and detailed style suitable for research discussion.
7. Avoid unsupported claims beyond the paths and their supporting evidence.
8. Make the answer self-contained rather than a visible trace of the generation process.

Output:
Return only the final answer.
```

Quality Audit Prompt

```
Assume you are an editor responsible for checking the validity, factuality, and consistency of a
generated protein-centric question-answer pair. The evidence contains raw context about the
protein and a sampled subgraph or reasoning path used as support. Treat the raw context and
sampled evidence as the only sources of truth.

Inputs:
- Question: {question}
- Answer: {answer}
- Evidence materials: {Information}

Audit criteria:
1. Protein-centricity and anonymization: the question should be about the given protein, and the
answer should address the same protein. The question should not reveal the protein name, but may
use "this protein" or "the given protein". The answer may contain the protein name if needed.
2. Validity: the answer must actually solve the protein-centric question. If the answer is
fragmented, off-topic, or fails to address the question, it is invalid.
3. Factuality: the QA pair must be supported by the raw context, including protein entry
information and literature evidence. If the QA contradicts the raw context, it is not factual.
4. Consistency: the QA pair must be consistent with the sampled subgraph or reasoning path, and
the sampled evidence must not contradict the raw context.
5. If the question or answer contains unsupported speculation against the supplied materials,
mark it as invalid.

Final output requirement:
Return Yes only if all criteria are satisfied; otherwise return No. Do not output additional
explanation.
```